

19 DEC 2005

10/561634

PCT/DE 2004/001234

BUNDESREPUBLIK DEUTSCHLAND

**PRIORITY
DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)



REC'D 30 JUL 2004

WIPO PCT

DE 04/1234

**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen: 103 27 549.5

Anmeldetag: 18. Juni 2003

Anmelder/Inhaber: Robert Bosch GmbH, 70469 Stuttgart/DE

Bezeichnung: Verfahren und Vorrichtung zur Fehlererkennung
für einen Cachespeicher und entsprechender
Cachespeicher

IPC: G 06 F, G 11 C

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der
ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 15. Juli 2004
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Stanschus

16.06.03 Sy/Ho

5

ROBERT BOSCH GMBH, 70442 Stuttgart

10

Verfahren und Vorrichtung zur Fehlererkennung für einen Cachespeicher und entsprechender Cachespeicher

Stand der Technik

15

Die Erfindung betrifft ein Verfahren und eine Vorrichtung zur Fehlererkennung für einen Cachespeicher sowie einen entsprechenden Cachespeicher gemäß den unabhängigen Ansprüchen.

20

Die Taktfrequenz heutiger Prozessoren liegt typischerweise deutlich höher als die Frequenz, mit der auf einen insbesondere externen Speicher zugegriffen werden kann. Um diese Zeitdifferenz auszugleichen, werden Cachespeicher eingesetzt. Durch Zusammenarbeit eines solchen schnellen Pufferspeichers mit einem entsprechenden Hauptspeicher können dann die Zugriffszeiten deutlich reduziert werden. Bei fehlertoleranten Rechnern ist es heutzutage üblich, dass bei dem Cachespeicher die Daten über einen fehlererkennenden oder fehlerkorrigierenden Code (ECC Error Code Correction) abgesichert werden. Treten aber in der Logik des Cachespeichers Fehler auf, reicht diese Datenabsicherung nicht mehr. Durch immer kleiner werdende Strukturen, insbesondere der Logik des Cachespeichers, wird diese immer anfälliger, insbesondere gegen transiente Fehler.

25

30

Daher ist es Aufgabe der Erfindung, neben der Datenabsicherung auch weitere Fehlermöglichkeiten auszuschließen und so einen sichereren Cachespeicher darzustellen, als es der Stand der Technik zu leisten vermag.

Vorteile der Erfindung

Die Erfindung zeigt ein Verfahren und eine Vorrichtung zur Fehlererkennung für einen Cachespeicher sowie einen entsprechenden Cachespeicher zur Speicherung von Daten, wobei der Zugriff auf die im Cachespeicher abgelegten Daten durch diesen zugeordnete Adressen erfolgt und vorteilhafter Weise zu den den abgelegten Daten zugeordneten Adressen wenigstens eine erste Prüfsignatur, bestehend aus wenigstens einem ersten Signaturbit erzeugt und ebenfalls im Cachespeicher abgelegt wird.

Dadurch ist es möglich, neben der bekannten Absicherung der Cachedaten auch die Cachelogik durch Fehlererkennung und/oder Fehlerkorrektur abzusichern, so dass ein Cachespeicher implementiert werden kann, bei dem bei einem Lesezugriff alle Fehler erkannt werden können, da eine Überprüfung der ersten Prüfsignatur bei jedem Lesezugriff auf den Cachespeicher erfolgreich durchführbar ist.

Zweckmäßiger Weise wird das wenigstens eine abgelegte erste Signaturbit der ersten Prüfsignatur mit einem zweiten Signaturbit einer zweiten Prüfsignatur, die aus einer am Cachespeicher angelegten Adresse gebildet oder bereits mit der Adresse mitübergeben (z.B. von der die CPU enthaltenden Recheneinheit) wird, verglichen.

Um zusätzlich Fehler im Gültigkeitsbit des Cachespeichers, also dem Validbit zu erkennen, wird dieses wenigstens doppelt abgelegt. Zweckmäßiger Weise wird das Gültigkeitsbit m -fach abgelegt und mittels eines n -aus- m -Tests überprüft, wobei m und n natürlich Zahlen sind, wobei m größer als 2 und n größer als $m/2$ ist. Dadurch kann mindestens Einfehlertoleranz erreicht werden.

Dabei kann das Gültigkeitsbit auch k -fach in Form eines 1 aus k Codes abgelegt werden, wobei k einer natürlichen Zahl entspricht, wodurch vorteilhafter Weise auch Mehrfachfehler erkannt werden können, da dabei nur 1 aus 2^k Bitkombinationen entsprechend dem 1 aus k Code gültig ist.

Besonders vorteilhaft ist der Einsatz der Erfindung bei einem Instruktionscache, also bei einem Cachespeicher, bei dem als Daten nur Instruktionen, sprich Befehle abgelegt werden. Da bei diesem nahezu ausschließlich Lesefehler zu berücksichtigen sind, kann dann das erfindungsgemäße Verfahren und die erfindungsgemäße Vorrichtung sowie ein

entsprechender Cachespeicher die Absicherung bzw. Fehlererkennung und Fehlerkorrektur nahezu komplett abdecken.

5 Vorteilhafter Weise werden Sets bzw. Gruppen von Daten mit den entsprechenden zugehörigen ersten Prüfsignaturen zusammen unter einem ersten Zeilenindex im Cachespeicher abgelegt, wobei dieser erste Zeilenindex aus dem Cachespeicher durch Zeilendecodierung wiedergewonnen wird und mit einem angelegten zweiten Zeilenindex auf Übereinstimmung verglichen wird.

10 Abhängig von dem jeweiligen Vergleich der Prüfsignaturen oder des Gültigkeitsbits oder des durch Zeilendecodierung rückgewonnenen Zeilenindex wird entsprechend ein Fehlersignal erzeugt. Dieses kann zum Einen generell erzeugt werden, also unterschiedlich bei erkanntem Fehler und bei erkannter Übereinstimmung, oder es wird lediglich bei erkanntem Fehler ein solches Fehlersignal erzeugt.

15 Zweckmäßiger Weise wird dieses Fehlersignal bei erkanntem Fehler dann als Cache-Miss-Signal behandelt, was zur Folge hat, dass solch fehlerhafte Daten bzw. Adressen nicht verwendet, sondern korrigiert oder ersetzt werden.

20 Werden vorteilhafter Weise bei Erkennung eines Fehlers die Daten bzw. Instruktionen erneut aus dem Hintergrundspeicher in den Cachespeicher geladen kann damit ein hoch verfügbares fehlertolerantes System erzielt werden, welches sich darüber hinaus durch kostengünstige Fehlererkennungsmechanismen auszeichnet.

25 Weitere Vorteile und vorteilhafte Ausgestaltungen ergeben sich aus der Beschreibung sowie den Merkmalen der Ansprüche.

Zeichnung

Im Weiteren wird die Erfindung anhand der in der Zeichnung dargestellten Figuren näher erläutert. Dabei zeigt

5 Figur 1 ein Beispiel für ein erfindungsgemäßes Cachespeichersystem und

Figur 2 zeigt symbolisch den Cachespeicher und die erfindungsgemäß vorgenommenen Vergleiche zur Fehlererkennung.

10

Ausführungsbeispiele

Figur 1 zeigt dabei einen Vordergrundspeicher oder Cachespeicher CS und einen Hauptspeicher oder Hintergrundspeicher HS. Weiterhin abgebildet ist ein

15 Zugriffsgenerator ZG, der beispielsweise einer Central Processing Unit entspricht. Zwischen die Einheit, die Daten anfordert bzw. ändert, also den Zugriffsgenerator ZG und den Hintergrundspeicher HS wird eine Steuerung, der sogenannte Cachecontroller CC geschaltet, welcher die Daten im Cachespeicher CS ablegt und so für weitere Zugriffe vorrätig hält. Der Zugriffsgenerator und der Cachecontroller sind über eine Adressleitung

20 ADR1, eine Datenleitung D1 und eine Schreib-/Leseleitung R/W miteinander verbunden. Ebenso ist der Cachecontroller CC mit dem Cachespeicher CS über eine Adressleitung ADR2, eine Datenleitung D2 sowie eine Schreib-/Leseleitung R/W verbunden. Solche Verbindungen bestehen auch zwischen Cachecontroller CC und Hauptspeicher HS mit Adressleitung ADR3, Datenleitung D3 und der Schreib-/Leseleitung R/W.

2

Ist nur eine Übertragungsrichtung notwendig, z. B. beim ausschließlichen Lesen von Befehlen aus dem Arbeitsspeicher bei einem Instruktionscache, also einem Cachespeicher, in dem nur Instruktionen abgelegt werden, können die R/W-Leitungen, also die Schreib-/Leseleitungen zwischen Zugriffsgenerator ZG und Cachecontroller CC

30 sowie zwischen Cachecontroller CC und Hintergrundspeicher HS entfallen. Allerdings muss der Cachecontroller CC auf den Vordergrundspeicher oder Cachespeicher CS immer schreiben und lesen können, so dass die R-/W-Leitung zwischen Cachecontroller CC und Cachespeicher CS auch bei einem Instruktionscache erhalten bleibt.

Zugriffswünsche bestehen somit aus einer Adresse und bei Schreibzugriffen den neuen Daten sowie bei einem nicht reinen Instruktionscache der Zugriffsart, also Lesen oder Schreiben übertragen über die R/W Read-Write-Leitung. Der Zugriffsgenerator ZG sendet solche Anforderungen oder Wünsche an den Cachecontroller CC, der diese bedient, und zwar durch geeignete Anforderungen an Cachespeicher CS und/oder Hintergrundspeicher HS. Handelt es sich bei dem Zugriffsgenerator um eine Central Processing Unit CPU oder ist dieser in der CPU enthalten, so steht die Zugriffsadresse im Allgemeinen in einem Speicheradressregister SAR und die zu schreibenden Daten in einem Schreibregister SR, wohingegen gelesene Daten im Allgemeinen in einem Leseregister LR aufgenommen werden.

Um die erfindungsgemäßen und nachfolgend noch zu erläuternden Vergleiche durchzuführen, ist eine Vergleicherkomponente VK vorgesehen. Zur Wiedergewinnung eines Zeilenindex ist ein Zeilendecodierer ZD vorgesehen, und für die Rückgewinnung der Adressinformation zum erfindungsgemäßen Vergleich ist ein Adressblock ADR vorgesehen. Je nach Anwendung kann eine unterschiedliche Anzahl von Vergleichen in der Vergleicherkomponente VK vorgesehen werden. Ebenso ist die Anwendung der Blöcke VK, ZD und ADR nicht fest und kann je nach Anwendung variieren, insbesondere können diese Blöcke vorteilhafter Weise in beliebiger Zusammensetzung auch extern zum Cachecontroller CC lokalisiert sein und mit diesem in Verbindung stehen.

Figur 2 zeigt nun erfindungsgemäße Methoden zur Fehlererkennung beim Cachespeicher, wobei die Grundvoraussetzung für den Einsatz dieses Cachespeichers die Absicherung der Adressen mit einem Fehlererkennungs- bzw. Fehlerkorrekturcode, also einer Signatur darstellt. Im einfachsten Fall kann diese Signatur nur aus einem Signaturbit, beispielsweise einem Paritybit bestehen. Andererseits kann die Absicherung auch durch komplexere ED-Codes (Error Detection) wie einen Berger-Code oder einen Bose-Lin-Code usw., oder auch durch einen komplexeren ECC-Code (Error Check and Correction) wie beispielsweise einen Hamming-Code, usw. realisiert werden, um durch entsprechende Bitzahl eine sicherere Fehlererkennung zu ermöglichen. Es kann aber auch als Codegenerator, beispielsweise eine Generatortabelle (fest verdrahtet oder in Software) verwendet werden, um bestimmten Eingangsmustern der Bits im Rahmen der Adresse ein gewünschtes Codemuster beliebiger Länge zuzuordnen.

In Figur 2 ist ein solcher Cachespeicher 100 symbolisch dargestellt. Darin sind die Bereiche 103, 104, 105, 106, 107, 108 und 110 unterschieden. In den Bereichen 105 und 107 und ff. auch in 110 usw. werden die jeweiligen Daten im Cachespeicher abgelegt, ergänzt durch eine ECC-Signatur, um Fehler in den im Cache abgelegten Speicher zu erkennen. Die Adresssignatur A-ECC ist hier in den Blöcken 106, 108 sowie 110 und ff. abgelegt und den entsprechenden Daten gemäß der Adresse zugeordnet. Im Beispiel besteht eine solche Adresse 111 aus einem Block 112, der einen Tag oder eine Kennung enthält, welches anzeigt, zu welcher Anfangsadresse der entsprechende Datenblock wirklich gehört, da, wie in Figur 2 dargestellt, mehrere Bereiche des Hintergrundspeichers siehe Daten 105 und 107 in eine Zeile bzw. einen Set abgebildet werden. In diesem Beispiel enthält gemäß Block 102 ein Set oder eine Zeile 8 Tags von Tag 0 bis Tag 7. Block 113 enthält den Zeilenindex, durch welchen die Zeilenanwahl, hier von Zeile 0 bis Zeile 7 gemäß Block 101 erfolgt. Weiterhin und optional sind Block 114 und Block 115 die der Wort- bzw. Byteanwahl innerhalb einer Zeile und eines Tags dienen. Die Felder 114 und 115 hängen stark von der verwendeten Wortbreite der einzelnen Datenfelder ab und sind nicht zwingend. Feld 116 enthält dann die Prüfsignatur der Adresse 111, also der Felder 112 bis 115.

Es gibt nun zwei grundsätzliche Arten von Fehlern im Cachespeicher: Zum Einen Fehler, in den im Cachespeicher abgelegten Daten und zum Anderen Fehler in der Cachelogik, insbesondere dem Cachecontroller. Daten, die in einem Cachespeicher sind, sind immer noch an einer anderen Stelle im Hauptspeicher bzw. Hintergrundspeicher oder sogar in einem nicht flüchtigen Speicher abgelegt. Wenn sichergestellt ist, dass die Daten in beiden Speichern konsistent sind, also Cachespeicher und Hintergrundspeicher müssen Fehler in den Daten im Cachespeicher folglich nur erkannt, aber nicht korrigiert werden. Dieses Erkennen kann durch Absicherung der Daten mit einem ED-Code oder einem ECC-Code, wie in Figur 2 dargestellt, geschehen. Sind die Daten nicht konsistent, z.B. bei einem Datencache kann ein ECC-Code wie in Figur 2 verwendet werden.

Beim Erkennen von Fehlern in der Cachelogik wird bei einem lesenden Zugriff auf ein Datenwort im Cachespeicher bei einem direkt zuordnenden Cache zuerst mit der Zeilenanwahl, also dem Zeilenindex, die entsprechende Cachespeicherzeile ausgewählt. Anschließend wird in dieser Zeile überprüft, ob das Tagfeld 112 in dieser Zeile stimmt entsprechend Block 102 und vgl. V4, in dem das Tagfeld über den Tagfeldvergleich VTag gemäß V4 mit dem Tag aus Block 102 verglichen wird und sich bei

Übereinstimmung ein Cachehit vergibt. Ist dies der Fall, also bei Cachehit, so wird bei einem gültigen Validbit V das entsprechende Wort aus der Zeile ausgelesen. Fehler in diesem Bereich der Cachelogik können in der Zeilenanwahl, dem Tagspeicher 102 und dem Komparator für das Tagfeld auftreten. Diese Fehler wirken sich darin aus, dass auf die falsche Zeile oder auf falsche Daten im Cachespeicher zugegriffen wird.

Um Fehler im Gültigkeitsbit, also dem Validbit zu erkennen, muss dies wenigstens dupliziert werden, wird also beispielsweise in Feld 103 oder in einem anderen Feld doppelt abgelegt. Vorteilhaft ist auch ein n-aus-m-Code, also ein n-aus-m-Test, so dass bei gleichartigem Kippen mehrerer Bits durch einen Kopplungsfehler immer noch eine Erkennung stattfinden kann, wobei n und m natürliche Zahlen sind, wobei $m > 2$ und $n > m/2$ ist. Wodurch wenigstens eine Einfehlertoleranz gewährleistet werden kann.

Dabei kann das Gültigkeitsbit auch k-fach in Form eines 1 aus k Codes abgelegt werden, wobei k einer natürlichen Zahl entspricht, wodurch vorteilhafter Weise auch Mehrfachfehler bei Bittkippen erkannt werden können, da dabei nur eine Bitkombination aus 2^k Bitkombinationen entsprechend dem 1 aus k Code gültig ist und somit durch Vergleich beliebige Abweichungen von der gültigen Bitkombination erkennbar sind.

Dabei gibt das Valid- oder Gültigkeitsbit an, ob in der Cachespeicherzelle überhaupt gültige Daten vorhanden sind, da sowohl das Tagfeld als auch der Datenblock immer ein Bitmuster enthalten. Die Daten entsprechen aber anfangs nicht dem und der Adresse stehenden Wert im Hintergrundspeicher. Auch hier können individuelle Flags für Teile des Blocks vorhanden sein. In Abschnitt 104 ist ein sogenanntes Verändert- oder Dirty-Flag bzw. Bitmuster D vorgesehen, welches angibt, ob der Inhalt des Datenblocks noch mit dem Hintergrundspeicher übereinstimmt oder gelesen und anschließend verändert wurde, aber noch nicht zurückgeschrieben ist.

So werden nun die Prüfsignaturen aus den Abschnitten 106 bzw. 108 usw., also A-ECC mit dem entsprechenden Prüfmuster, also der Prüfsignatur entsprechend Block 116 gemäß Pfeil V-ECC und V2 verglichen und das Ergebnis dann entsprechend V3 mit dem korrekten geprüften Validbit überprüft, um dann bei Übereinstimmung die Aussage "gültige Daten" zu erhalten.

Die Zeilenanwahl lässt sich erfindungsgemäß als Zeilendecoder oder Zeilendecodierung betrachten. Fehler werden hier durch Zurückdecodieren der ausgewählten Zeile und Vergleich der daraus gewonnenen Adresse mit dem angewählten Zeilenindex erkannt, wie dies über Vergleich V1 dargestellt ist.

5

Die zusätzlich zu den in einem gängigen Cachespeicher vorgesehenen Komparatoren, also Vergleicher, hier symbolisch mit V1 bis V4 bezeichnet, können in der in Figur 1 dargestellten Vergleichskomponente VK gesammelt untergebracht sein. Die Bildung des jeweiligen Adress-ECC erfolgt im Block ADR gemäß Figur 1, und die Zeilendecodierung im Block ZD. Die Bildung des jeweiligen Adress-ECC kann auch in der CPU der Recheneinheit also insbesondere in ZG erfolgen.

10

Bei der Überprüfung des Tagfelds und eventuell auftretenden Fehlern geschieht der Vergleich einer besonderen Ausführungsform bitweise. Dann kann also im Fehlerfall von einem Einfachfehler ausgegangen werden. Dieser Einfachfehler wird dann durch den Fehlererkennungscode der Adresse A-ECC, der mit den entsprechenden Daten abgelegt ist, erkannt. Somit können dann alle Fehler, eben auch transiente Fehler, im Cachespeicher erkannt werden. Dies gilt für den beschriebenen Einfachfehlerfall selbst dann, wenn für die Adressabsicherung lediglich ein Prüfbit, insbesondere ein Paritybit verwendet wird. Fehler in der Wortauswahl werden ebenfalls durch Vergleich der mitgespeicherten Prüfsignatur der Adresse erkannt.

15

20

Das erfindungsgemäße Verfahren, die Vorrichtung und der entsprechende Cachespeicher kann auch eingesetzt werden, um bestimmte Fehler beim Schreiben auf einen Cachespeicher zu selektieren. Tritt bei einem Schreiben in den Cachespeicher durch einen Fehler des Cachecontrollers CC ein Adressierungsfehler auf, so kann die falsche Zeilenadresse, also der falsche Zeilenindex ebenfalls durch den Zeilendecodierer erkannt werden. Wird das Tagfeld verfälscht, so wird der Fehler durch den mit den Daten gespeicherten Fehlererkennungscode beim Wiederauslesen der Daten erkannt. Ebenso werden Fehler in den Adresscodierungsbits ebenfalls beim Auslesen erkannt.

25

30

Bei einem reinen Instruktionscache, also einem Cachespeicher, in dem nur Befehle als Daten abgelegt sind, können fast ausschließlich Lesefehler berücksichtigt werden, so dass dann ein Fehler durch die Fehlererkennung beim Auslesen sofort erkannt wird.

35

Werden bei Erkennung eines Fehlers die Daten bzw. Instruktionen erneut aus dem Hintergrundspeicher in den Cachespeicher geladen kann damit ein hoch verfügbares fehlertolerantes System erzielt werden, welches sich darüber hinaus durch kostengünstige Fehlererkennungsmechanismen auszeichnet. Dabei kann ein solches wiederholtes Laden aus dem Hintergrundspeicher von zwei Faktoren abhängig gemacht werden und zwar zum einen vom Fehler bzw. der jeweiligen Fehlererkennung und zum anderen vom Daten- bzw. Instruktionsabschnitt in dem der Fehler auftrat. D.h. im einfachsten Fall werden bei Auftritt eines Fehlers die Daten (also Daten und/oder Instruktionen) komplett erneut aus dem Hintergrundspeicher in den Cachespeicher geladen. Abhängig von bestimmten Fehlern entsprechend der vorgenannten Fehlererkennungsmechanismen können auch nur Teile des zuletzt geladenen Datenbestandes erneut geladen werden (z.B. je gravierender der Fehler desto größere Datenanteile). Abhängig von diesen beiden Faktoren sind dann beliebige Abstufungen und Beziehungen möglich.

Damit kann entsprechend der erfindungsgemäßen Verfahren und Vorrichtungen sowie Cachespeichern ein sicherer und hochverfügbarer Cachespeicher angegeben werden, der auf kostengünstige Weise sowohl gegen Fehler in den Daten als auch gegen Fehler in der Cachelogik abgesichert ist.

16.06.03 Sy/Ho

5

ROBERT BOSCH GMBH, 70442 Stuttgart

10

Patentansprüche

15

1. Verfahren zur Fehlererkennung bei einem Cachespeicher zur Speicherung von Daten, wobei der Zugriff auf die im Cachespeicher abgelegten Daten durch diesen zugeordnete Adressen erfolgt, dadurch gekennzeichnet, dass zu den, den abgelegten Daten zugeordneten Adressen, wenigstens eine erste Prüfsignatur bestehend aus wenigstens einem ersten Signaturbit erzeugt und ebenfalls im Cachespeicher abgelegt wird.

20

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass eine Überprüfung der ersten Prüfsignatur bei jedem Lesezugriff auf den Cachespeicher erfolgt.

25

3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass das wenigstens eine abgelegte erste Signaturbit der ersten Prüfsignatur mit einem zweiten Signaturbit einer zweiten Prüfsignatur, die aus einer am Cachespeicher angelegten Adresse gebildet wird verglichen wird.

30

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass das wenigstens eine abgelegte erste Signaturbit der ersten Prüfsignatur mit einem zweiten Signaturbit einer zweiten Prüfsignatur verglichen wird, die mit einer am Cachespeicher angelegten Adresse mitübergeben wird.

5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass ein Gültigkeitsbit im Cachespeicher wenigstens doppelt abgelegt wird.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass das Gültigkeitsbit m-fach abgelegt wird und mittels n aus m Test überprüft wird, wobei m, n natürliche Zahlen sind und $m > 2$ und $m > n > m/2$.

5 7. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass das Gültigkeitsbit k-fach als Bitkombination in Form eines 1 aus k Codes abgelegt wird, wobei durch Vergleich 1 Bitkombination aus 2^k Bitkombinationen als gültig erkannt wird, wobei k einer natürlichen Zahl entspricht.

10 8. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Daten nur aus Instruktionen bestehen.

15 9. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass einer Gruppe von Daten und den zugehörigen ersten Prüfsignaturen jeweils ein erster Zeilenindex im Cachespeicher zugeordnet wird und mit einem zweiten am Cachespeicher angelegten Zeilenindex verglichen wird, wobei dieser erste Zeilenindex durch Zeilendecodierung aus dem Cachespeicher wiedergewonnen wird.

20 10. Verfahren nach Anspruch 3 oder 4 oder 5 oder 6 oder 7 oder 9, dadurch gekennzeichnet, dass abhängig vom jeweiligen Vergleichsergebnis ein Fehlersignal erzeugt wird.

25 11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, dass das Fehlersignal als Cache-Miss-Signal behandelt wird.

12. Verfahren nach Anspruch einem der Ansprüche 2 bis 9, dadurch gekennzeichnet, dass abhängig vom jeweiligen Vergleichsergebnis wenigstens ein Fehler erkannt wird und bei Erkennung eines Fehlers die Daten erneut in den Cachespeicher geladen werden.

30 13. Vorrichtung zur Fehlererkennung bei einem Cachespeicher zur Speicherung von Daten, wobei der Zugriff auf die im Cachespeicher abgelegten Daten durch diesen zugeordnete Adressen erfolgt, dadurch gekennzeichnet, dass Mittel enthalten sind, die zu den, den abgelegten Daten zugeordneten Adressen wenigstens eine erste Prüfsignatur bestehend

aus wenigstens einem ersten Signaturbit erzeugen und ebenfalls im Cachespeicher ablegen.

14. Cachespeicher mit einer Vorrichtung zur Fehlererkennung, wobei der Zugriff auf die im Cachespeicher abgelegten Daten durch diesen zugeordnete Adressen erfolgt, dadurch gekennzeichnet, dass zu den, den abgelegten Daten zugeordneten Adressen wenigstens eine erste Prüfsignatur bestehend aus wenigstens einem ersten Signaturbit erzeugt wird und diese erste Prüfsignatur im Cachespeicher abgelegt ist.

16.06.03 Sy/Ho

5

ROBERT BOSCH GMBH, 70442 Stuttgart

10

Verfahren und Vorrichtung zur Fehlererkennung für einen Cachespeicher und entsprechender Cachespeicher

Zusammenfassung

15

Verfahren zur Fehlererkennung bei einem Cachespeicher zur Speicherung von Daten, wobei der Zugriff auf die im Cachespeicher abgelegten Daten durch diesen zugeordnete Adressen erfolgt, dadurch gekennzeichnet, dass zu den, den abgelegten Daten zugeordneten Adressen wenigstens eine erste Prüfsignatur bestehend aus wenigstens einem ersten Signaturbit erzeugt und ebenfalls im Cachespeicher abgelegt wird.

20

(Figur 1)

R 30607J

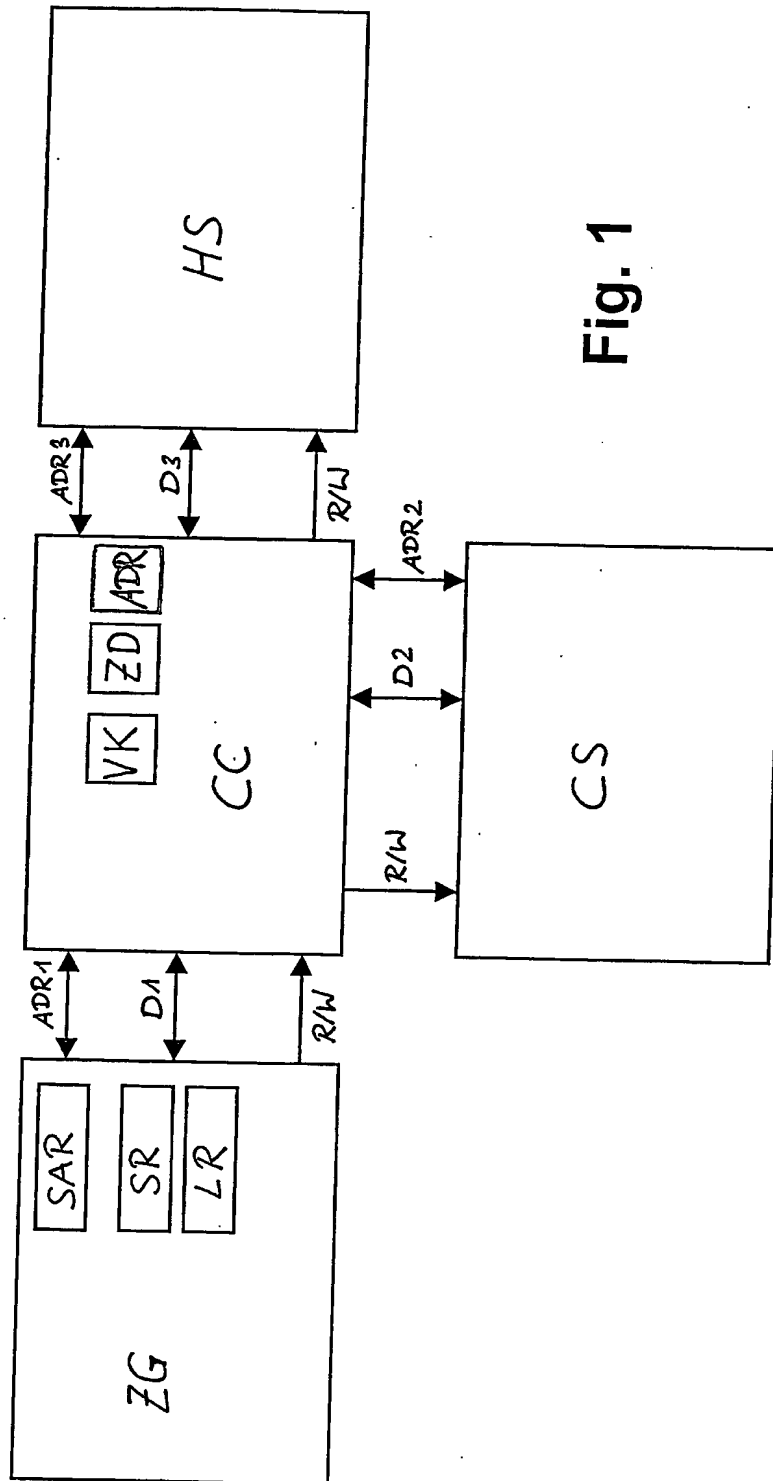


Fig. 1

[illegible]

25